Error-Bounded Approximations for Infinite-Horizon Discounted Decentralized POMDPs

Jilles Steeve Dibangoye, Olivier Buffet, François Charpillet

Error-bounded Approximations for Infinite-Horizon Discounted Decentralized POMDPs

Jilles S. Dibangoye, Olivier Buffet, and François Charpillet

Inria & Université de Lorraine
Nancy, France
firstname.lastname@inria.fr

Abstract. We address decentralized stochastic control problems represented as decentralized partially observable Markov decision processes (Dec-POMDPs). This formalism provides a general model for decision-making under uncertainty in cooperative, decentralized settings, but the worst-case complexity makes it difficult to solve optimally (NEXP-complete). Recent advances suggest recasting Dec-POMDPs into continuous-state and deterministic MDPs. In this form, however, states and actions are embedded into high-dimensional spaces, making accurate estimate of states and greedy selection of actions intractable for all but trivial-sized problems. The primary contribution of this paper is the first framework for error-monitoring during approximate estimation of states and selection of actions. Such a framework permits us to convert state-of-the-art exact methods into error-bounded algorithms, which results in a scalability increase as demonstrated by experiments over problems of unprecedented sizes.

Keywords: decentralized stochastic control, error-bounded approximations.

Learning and planning algorithms for decentralized stochastic control problems are of importance in a number of practical domains such as network communications and control; rescue, surveillance and exploration tasks; multi-robotics; collaborative games [15]; to cite a few. Decentralized partially observable Markov decision processes (Dec-POMDPs) have emerged as a standard framework for modeling and solving such problems [6]. This formalism involves a set of agents with different, but related, observations about the world, which cooperate to achieve a common long-term goal, but cannot explicitly communicate with one another. While many decentralized stochastic control problems can be formalized as Dec-POMDPs, only a few of them can be solved optimally due to their worst-case complexity: finite horizon problems are in NEXP, and infinite horizon problems are undecidable [6]. This intractability is due to the doubly-exponential growth in required computational resources, making it hard to find an optimal solution for all but the smallest instances [12,5].

A recent scalability increase builds upon two fundamental results [7]. The first result establishes that Dec-POMDPs can be transformed with no loss of optimality into continuous-state and deterministic MDPs, called occupancy MDPs. In this form, the states —called occupancy states— are distributions over the states and action-observation joint histories of the original Dec-POMDPs, and the actions —called decentralized decision rules— are mappings from joint histories to joint actions of the

original Dec-POMDPs. Secondly, the optimal value function of a finite-horizon occupancy MDP is a piecewise linear and convex function of the occupancy state. These results allow to combine advances in continuous-state MDP and POMDP algorithms, which (among others) result in the feature-based heuristic search value iteration algorithm (FB-HSVI). This algorithm can produce optimal solutions for medium-sized problems and medium planning horizons, but quickly runs out of time and memory for larger-scale problems and planning horizons. Such limited scalability is mainly because states and actions of occupancy MDPs are embedded into high-dimensional spaces, making accurate estimate of states and greedy selection of actions intractable for all but trivial-sized problems.

A natural question to ask is whether approximate (error-bounded) solutions can be found efficiently for decentralized stochastic control problems. On the one hand, memory-bounded dynamic programming algorithms for solving infinite-horizon discounted Dec-POMDPs are often quite effective at finding good heuristic solutions, while requiring bounded computational resources [20,9,13,10]. However, these methods do not come with rigorous guarantees concerning the quality of the final heuristic solution. On the other hand, error-bounded algorithms for solving infinite-horizon discounted Dec-POMDPs exist. Examples include error-bounded methods for discounted POMDPs that are (or can be) transferred back to discounted Dec-POMDPs: policy iteration (PI) [5]; incremental policy generation (IPG) [2]; point-based value iteration (PBVI) [14,17]; and heuristic search value iteration (HSVI) [7,21]. These algorithms rely either on ϵ-pruning methods[1] (PI and IPG) or/and on exploration strategies that focus on a small subset of the search space (PBVI and HSVI), but they all make use of greedy action-selection and accurate state-estimation operators, which quickly exhausts the available resources before convergence. Furthermore, theoretical analyses of point-based approaches (e.g., PBVI) demonstrate that resulting error bounds are loose and have only a theoretical significance [21,10].

In this paper, we focus on characterizing efficient error-bounded solutions for infinite-horizon discounted decentralized stochastic control problems. The novel approach proceeds by converting infinite-horizon discounted Dec-POMDPs into finite-horizon discounted occupancy MDPs, thereby computing a non-stationary policy over a finite planning horizon. In such a setting, approximations are typically achieved by replacing greedy action-selection and accurate state-estimation operators by approximate counterparts. In addition, we preserve the ability to bound the error with respect to the optimal infinite-horizon value function. Our study differs from previous studies in that it directly bounds the regret, avoiding the max-norm machinery of previous analyses of value and policy iteration algorithms [18,4,17,21,10,19], which may result in tighter bounds. We further extend the state-of-the-art feature-based heuristic search value iteration algorithm to incorporate the error we make in both greedy action-selection and accurate state-estimation. The result is an algorithm that can solve problems of unprecedented sizes from the literature while providing strong theoretical guarantees.

In the remainder of this paper, we will introduce in Section 1 the Dec-POMDP framework and the reformulation into occupancy MDPs. Section 2 extends, from finite-

[1] An ϵ-pruning method circumvents regions of the search space that cannot significantly improve the current solution, and the resulting solution is guaranteed to be at ϵ of the optimum.

horizon settings to infinite-horizon ones, recent advances in optimally solving Dec-POMDPs as occupancy MDPs. Then, we describe the novel approximation framework, derive theoretical guarantees and algorithmic extensions in Section 3. Finally, we present in Section 4 experimental results demonstrating the scalability of the resulting error-bounded algorithm.

1 From Dec-POMDPs to OMDPs

This section presents formalisms for the infinite-horizon discounted decentralized partially observable Markov decision process (Dec-POMDP) and its associated occupancy Markov decision process (OMDP).

1.1 Decentralized stochastic control problems as Dec-POMDPs

The Dec-POMDP framework formalizes a discrete stochastic system that evolves under the influence of N agents. A key assumption in this framework is that agents cannot directly observe the true state of the system. In fact, they have different but related observations about the state of the system and cannot explicitly communicate with one another. Nevertheless, they need to cooperate in order to achieve a common long-term goal, i.e., to select actions that maximize the collection of rewards in the long run.

Definition 1 (Dec-POMDP). *An N-agent decentralized partially observable Markov decision process is given as a tuple $M \equiv (S, \{A^i\}, \{Z^i\}, p, r, b_0, \gamma)$, where: S is a finite set of hidden states; A^i is a finite set of private actions of agent $i \in \{1, 2, \ldots, N\}$; Z^i is a finite set of private observations of agent $i \in \{1, 2, \ldots, N\}$; $p^{a,z}(s, s') = Pr(s', z|s, a)$ is a dynamics model of the team of agents as a whole; $r(s, a)$ is a reward model of the team of agents as a whole; b_0 is an initial belief state; and $\gamma \in (0, 1)$ is a discount factor.*

The goal of solving M is to find an N-tuple of private policies $\pi \equiv (\pi^i)_{i\in\{1,2,\ldots,N\}}$ that yields the highest discounted total reward starting at b_0:

$$V^{\pi}_{M,\gamma,0}(b_0) = \boldsymbol{E}\left\{\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t, a_t) \mid \pi, b_0\right\}. \tag{1}$$

Let decentralized policy π be a N-tuple of private policies $(\pi^i)_{i\in\{1,2,\ldots,N\}}$. Each private policy π^i is a sequence of private decision rules $(\pi^i_t)_{t\in\{0,1,\ldots,\infty\}}$. The t-th private decision rule $\pi^i_t \colon \Theta^i_t \mapsto A^i$ of agent i prescribes private actions based on the whole information available to the agent up to time step t, namely its complete history of past actions and observations $\theta^i_t = (a^i_0, z^i_1, \ldots, a^i_{t-1}, z^i_t) \in \Theta^i_t$. We define Θ^i_t to be the set of all length-t private histories of actions and observations agent i may have experienced, $\Theta_t \equiv \times_{i\in\{1,2,\ldots,N\}}\Theta^i_t$ the set of joint histories and $\Theta = \cup_{t\in\{0,1,\ldots,T-1\}}\Theta_t$. In addition, we define the t-th decentralized decision rule π_t to be an N-tuple $(\pi^i_t)_{i\in\{1,2,\ldots,N\}}$ of private decision rules.

Since the history length grows as time goes on, for infinite horizon cases, this would require private decision rules to have infinite memory, which is not possible in practice. Therefore, we shall specify the nature of the decentralized policies we target in more detail. We first notice that the optimal value function over an infinite horizon can be arbitrarily accurately approximated by the optimal value function over a finite horizon.

To this end, we choose finite horizon T so that the regret of operating only over $T = \lceil \log_\gamma ((1-\gamma)\varepsilon/\|r\|_\infty) \rceil$ steps instead of an infinite number of steps is upper-bounded by any arbitrarily small scalar $\varepsilon > 0$, where $\|r\|_\infty = \max\{|r(s,a)| : \forall s \in S, \forall a \in A\}$. Indeed, the regret is upper-bounded by the cumulated sum of discounted losses from time step T onwards, so that: $\sum_{t=T}^{\infty} \gamma^t \|r\|_\infty \leq \varepsilon$.

In the remainder of this paper, we restrict the search space to decentralized policies described over planning horizon T. Unlike infinite-horizon decentralized policies, finite-horizon decentralized policies require a finite memory. At the execution phase, agents follow actions their private policies prescribe up to time step T; thereafter they behave randomly. Doing so, we are guaranteed to achieve performance with bounded error as discussed later below. Before proceeding any further, we next consider a reformulation of finite-horizon Dec-POMDPs into occupancy MDPs.

1.2 Occupancy Markov decision processes

The decentralized partially observable Markov decision process framework formalizes a decentralized stochastic control problem from a *perspective oriented towards agents*. In such a setting, agents are unaware of which actions the other agents take and which observations they receive; each agent behavior is based only upon its private histories. In this section, however, we formalize decentralized stochastic control problems from a *perspective oriented towards centralized solution methods*. In such a perspective, the system evolves under the control of agents based upon the total information about the state of the system the centralized solution method makes available to all agents prior to the execution phase, namely the information state.

The t-th information state $\zeta_t \equiv (b_0, \pi_0, \ldots, \pi_{t-1})$ is a sequence of decentralized decision rules starting at initial belief state b_0. It satisfies the following recursion: $\zeta_0 \equiv (b_0)$ and $\zeta_t \equiv (\zeta_{t-1}, \pi_{t-1})$, for all $t \in \{1, \ldots, T-1\}$. Next, it will prove useful to introduce the concept of occupancy states, as a means of maintaining a concise representation of the information state. A t-th occupancy state ξ_t is a distribution $P^{\zeta_t}(s_t, \theta_t)$ over histories and hidden states of M conditional on an information state ζ_t. For the sake of simplicity, we use notation $\Theta(\xi_t)$ to represent histories that are reachable in occupancy state ξ_t. The occupancy state has many important properties. First, it is a sufficient statistic of the information state when estimating the (current and future) reward to be gained by executing a decentralized decision rule: $\boldsymbol{R}(\xi_t, \pi_t) = \sum_{s_t} \sum_{\theta_t} \xi_t(s_t, \theta_t) \cdot r(s_t, \pi_t(\theta_t))$. In addition, it describes a deterministic and Markov decision process, where next occupancy state $\xi_{t+1} \equiv \boldsymbol{P}(\xi_t, \pi_t)$ depends only upon the current occupancy state ξ_t and the next decentralized decision rule π_t:

$$\xi_{t+1}(s', (\theta_t, a_t, z_{t+1})) = \mathbf{1}_{\{a_t\}}(\pi_t(\theta_t)) \sum_{s \in S} \xi_t(s, \theta_t) \cdot p^{a_t, z_{t+1}}(s, s'), \tag{2}$$

for $s' \in S$, $a_t \in A$, $z_{t+1} \in Z$, $\theta_t \in \Theta$ and where $\mathbf{1}_F$ is an indicator function. This process is known as the occupancy Markov decision process.

Definition 2 (OMDP). *Let $\hat{M} \equiv (\triangle, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P}, \gamma, \xi_0, T)$ be the T-steps OMDP with respect to Dec-POMDP M, where γ is a discount factor; ξ_0 corresponds to the initial belief in M; $\triangle \equiv \cup_{t \in \{0,1,\ldots,T\}} \triangle_t$ is the set of occupancy states up to time T; $\boldsymbol{A} \equiv \cup_{t \in \{0,1,\ldots,T\}} \boldsymbol{A}_t$*

is the finite set of decentralized decision rules; $\boldsymbol{R}(\xi_t, \pi_t)$ *is the reward model;* $\boldsymbol{P}(\xi_t, \pi_t)$ *is the transition rule; and* T *is a planning horizon.*

It is worth noticing that OMDP $\hat{M}$ can be seen as a generative model for occupancy states $\boldsymbol{P}(\xi_t, \pi_t)$ and rewards $\boldsymbol{R}(\xi_t, \pi_t)$, for all time step $t \in \{0, 1, \ldots, T-1\}$. A recent result shows that an optimal solution for $\hat{M}$, together with the correct estimation of the occupancy states, will give rise to the optimal solution of the original Dec-POMDP M over finite horizon T [8].

2 Optimally Solving Dec-POMDPs as OMDPs

This section reviews how to optimally solve Dec-POMDPs as OMDPs, a theory originally introduced under the total reward criterion [7,8]. Here, we extend it to deal with the discounted total reward criterion.

2.1 Bellman's optimality equations

In this subsection, we extend dynamic programming properties, including Bellman's optimality equations, to OMDPs (respectively Dec-POMDPs). Before proceeding any further, we start with preliminary definitions.

The discounted total reward of a decentralized policy $\pi \equiv (\pi_t)_{t\in\{0,1,\ldots,T-1\}}$ over T time steps and starting at occupancy state ξ_t is

$$V^{\pi}_{\hat{M},\gamma,t}(\xi_t) = \left[\sum_{k=t}^{T-1} \gamma^{k-t} \boldsymbol{R}(\xi_k, \pi_k) \mid \xi_{k+1} = \boldsymbol{P}(\xi_k, \pi_k)\right], \tag{3}$$

where the occupancy state sequence $(\xi_k)_{k\in\{t,t+1,\ldots,T-1\}}$ is generated by the deterministic transition rule $\boldsymbol{P}$ under decentralized policy π: $\xi_{k+1} = \boldsymbol{P}(\xi_k, \pi_k)$, $\forall k \in \{t, t+1, \ldots, T-1\}$ and $\forall t \in \{0, 1, \ldots, T-1\}$. Therefore, the optimal value function starting at occupancy state ξ_0 is $V^{*}_{\hat{M},\gamma,0}(\xi_0) = \max_{\pi} V^{\pi}_{\hat{M},\gamma,0}(\xi_0)$. Hence, the optimal value function $(V^{*}_{\hat{M},\gamma,t})_{t\in\{0,1,\ldots,T\}}$ is a solution of Bellman's optimality equation for $\hat{M}$, given by:

$$V^{*}_{\hat{M},\gamma,t}(\xi_t) = \max_{\pi_t \in \boldsymbol{A}_t} \left\{\boldsymbol{R}(\xi_t, \pi_t) + \gamma V^{*}_{\hat{M},\gamma,t+1}(\boldsymbol{P}(\xi_t, \pi_t))\right\}, \quad \forall \xi_t \in \triangle \tag{4}$$

and for $t = T$, we add a boundary condition $V^{*}_{\hat{M},\gamma,T}(\cdot) = 0$. If it can be solved for $(V^{*}_{\hat{M},\gamma,t})_{t\in\{0,1,\ldots,T\}}$, an optimal decentralized policy $\pi^{*} \equiv (\pi^{*}_t)_{t\in\{0,1,\ldots,T-1\}}$ may typically be obtained by maximization of the right-hand side for each ξ_t, i.e.,

$$\pi^{*}_t \in \arg\max_{\pi_t \in \boldsymbol{A}_t} \left\{\boldsymbol{R}(\xi_t, \pi_t) + \gamma V^{*}_{\hat{M},\gamma,t+1}(\boldsymbol{P}(\xi_t, \pi_t))\right\}, \quad \forall \xi_t \in \triangle. \tag{5}$$

2.2 Dynamic programming update operators

This subsection formally introduces the dynamic programming update operators involved in solving OMDPs, including: Bayesian state estimation; Bellman's evaluation and backup operators; and greedy action selection. To better understand this, let $\mathcal{V}$ be the set of real-valued functions $f\colon \triangle_t \mapsto \mathbb{R}$ for all $t \in \{0, 1, \ldots, T\}$.

Definition 3 (Bellman's evaluation operator). *For each decentralized decision rule $\pi_t \in \boldsymbol{A}_t$, let $\boldsymbol{T}_{\pi_t} : \mathcal{V} \mapsto \mathcal{V}$ be Bellman's evaluation operator, given by:*

$$(\boldsymbol{T}_{\pi_t} V_{\hat{M},\gamma,t+1})(\xi_t) = \boldsymbol{R}(\xi_t, \pi_t) + \gamma V_{\hat{M},\gamma,t+1}(\boldsymbol{P}(\xi_t, \pi_t)), \quad \forall \xi_t \in \triangle, \pi_t \in \boldsymbol{A}_t. \tag{6}$$

Bellman's evaluation operator transforms any arbitrary value function into a new value function based on a specified decentralized decision rule. It is worth noticing that Bellman's optimality equations (Equation 4) and greedy decision rule selections (Equation 5) can be stated in terms of the expression depending on occupancy state, decentralized decision rule and Bellman's evaluation operator. In the following, we formally define greedy selection and Bellman's update operators.

Definition 4 (Greedy action-selection operator). *For each decentralized decision rule $\pi_t \in \boldsymbol{A}_t$, let $\boldsymbol{G} \colon \mathcal{V} \mapsto (\triangle \mapsto \boldsymbol{A})$ be the greedy operator, given by:*

$$(\boldsymbol{G} V_{\hat{M},\gamma,t+1})(\xi_t) = \arg\max_{\pi_t \in \boldsymbol{A}_t} (\boldsymbol{T}_{\pi_t} V_{\hat{M},\gamma,t+1})(\xi_t), \quad \forall \xi_t \in \triangle, V_{\hat{M},\gamma,t+1} \in \mathcal{V}. \tag{7}$$

Together the greedy action-selection and Bellman's evaluation operators permit us to define Bellman's update operator as follows.

Definition 5 (Bellman's update operator). *Let $\boldsymbol{T} \colon \mathcal{V} \mapsto \mathcal{V}$ be Bellman's update operator, given by:*

$$(\boldsymbol{T} V_{\hat{M},\gamma,t+1})(\xi_t) = (\boldsymbol{T}_{(\boldsymbol{G} V_{\hat{M},\gamma,t+1})(\xi_t)} V_{\hat{M},\gamma,t+1})(\xi_t), \quad \forall \xi_t \in \triangle, V_{\hat{M},\gamma,t+1} \in \mathcal{V}. \tag{8}$$

Bellman's update operator maintains the value of a given occupancy state based on the greedy decentralized decision rule for a specified value function. When optimized exactly, the value function, solution of Bellman's optimality equations (Equation 4), is a *piecewise-linear and convex* function of the occupancy states [8]. That is, there exist finite sets of hyperplanes $(\Lambda_t)_{t \in \{0,1,\ldots,T-1\}}$, such that: $V^*_{\hat{M},\gamma,t}(\xi_t) = \max_{\lambda_t \in \Lambda_t} \sum_{s_t,\theta_t} \lambda_t(s_t, \theta_t) \cdot \xi_t(s_t, \theta_t)$, where $\lambda_t \in \mathbb{R}^{|S||\Theta_t|}$ for all $t \in \{0, 1, \ldots, T-1\}$.

Mappings $\boldsymbol{G}$ and $\boldsymbol{T}$ serve to define a dynamic programming methodology for the solution of occupancy Markov decision process $\hat{M}$. In particular, the piecewise-linearity and convexity property of the value function, together with mappings $\boldsymbol{G}$ and $\boldsymbol{T}$, allow to combine advances in continuous-state MDP and POMDP algorithms, which have led to the development of a novel family of exact algorithms, including the *feature-based heuristic search value iteration* [7,8].

2.3 The feature-based heuristic search value iteration

This subsection provides a succinct description of feature-based heuristic search value iteration (FB-HSVI) (Algorithm 1), which was originally introduced under the total reward criterion. Here, we extend it to address the discounted total reward criterion and discuss complexity issues.

The FB-HSVI algorithm's description. FB-HSVI extends to decentralized stochastic control problems the *heuristic search value iteration* (HSVI) algorithm, which was originally developed for partially observable Markov decision processes [21]. Similarly to HSVI, it corresponds to a family of trial-based algorithms that searches an optimal solution of an occupancy Markov decision process. FB-HSVI proceeds by generating trajectories of occupancy states, starting at the initial occupancy state. It maintains both upper and lower bounds over the optimal value function. It guides exploration towards occupancy states that are more relevant to the upper bound by greedily selecting decentralized decision rules with respect to the upper bound, and reducing the gap between bounds at visited occupancy states. If the gap between upper and lower bounds at the initial occupancy state is ε, then it terminates. In such a case, we are guaranteed FB-HSVI has converged to an ε-optimal solution, as initially targeted. Though FB-HSVI is already equipped with a mechanism for finding ε-optimal solutions —since it uses greedy action-selection and accurate state-estimation operators— in practice it quickly exhausts the available resources before convergence. To better understand this, we provide a complexity analysis of each operator involved in FB-HSVI.

Algorithm 1: The feature-based heuristic search value iteration for $\hat{M}$ (resp. M)

1 **function** FB-HSVI($\hat{M}, \varepsilon, (\underline{V}_{\hat{M},\gamma,t})_{t\in\{0,1,\cdots,T\}}, (\bar{V}_{\hat{M},\gamma,t})_{t\in\{0,1,\cdots,T\}}$)
2 ⌊ **while** GAP(ξ_0) $> \varepsilon$ **do** EXPLORE (ξ_0)

3 **function** GAP(ξ_t)
4 ⌊ **return** $\bar{V}_{\hat{M},\gamma,t}(\xi_t) - \underline{V}_{\hat{M},\gamma,t}(\xi_t)$

5 **function** EXPLORE (ξ_t)
6 ⌊ **if** GAP(ξ_t) $> \varepsilon/\gamma^t$ **then**
7 ⌊ ⌊ $\pi_t^* \leftarrow (\boldsymbol{G}\bar{V}_{\hat{M},\gamma,t+1})(\xi_t)$
8 ⌊ ⌊ EXPLORE($\boldsymbol{P}(\xi_t, \pi_t^*)$)
9 ⌊ ⌊ $(\boldsymbol{T}\bar{V}_{\hat{M},\gamma,t+1})(\xi_t)$ and $(\boldsymbol{T}\underline{V}_{\hat{M},\gamma,t+1})(\xi_t)$

Complexity of dynamic programming operators. As FB-HSVI proceeds, there are three operations that can significantly affect the overall performance: the greedy action-selection operator $\boldsymbol{G}$; the accurate state-estimation operator $\boldsymbol{P}$; and finally, Bellman's update operator $\boldsymbol{T}$. To better understand the complexity involved in these operations, let $|V|$ be the size of value function V (respectively the upper- or lower-bound value functions). Let $\Theta^i(\xi_t)$ be the set of private histories of agent i involved in occupancy state ξ_t, $|\Theta^*(\xi_t)| = \max_{i\in 1,2,\ldots,N} |\Theta^i(\xi_t)|$ and $|A^*| = \max_{i\in 1,2,\ldots,N} |A^i|$. Algorithm 1 (lines 7 and 9) performs a greedy action-selection operator $\boldsymbol{G}$, which involves enumerating and evaluating exponentially many decentralized decision rules in the worst case, and requires time complexity $O(|V|^{|\Theta^*(\xi_t)|^{|N|A^*|}})$ that grows doubly exponentially with increasing number of private histories involved in the occupancy state ξ_t. In practice, branch-and-bound methods explore only a small portion of this set, which saves considerable time [7,8]. Then, Algorithm 1 (line 8) computes the next occupancy state given the current one and the next decentralized decision rule. Unlike the greedy action-selection operator, this state-estimation rule has complexity $O(|S|^2|\Theta(\xi_t)||Z|)$, that is polynomial in the number of joint histories involved in the occupancy states and the number of joint observations.

However, in the worst case, the number of joint histories increases by a factor of $|Z|$ as time goes on. This may limit ability to perform the greedy action-selection operator later on. Finally, Algorithm 1 (line 9) performs Bellman's update operator $\boldsymbol{T}$ to maintain both upper and lower bounds at a given occupancy state, namely *point-based Bellman's update*. Unlike the full Bellman's update operator, the point-based Bellman's update operator maintains the value function only at a single occupancy state at a time, which makes it significantly more tractable. Nonetheless, the complexity of this operation remains time demanding as it requires performing a greedy action selection.

Given that the complexity of operators $\boldsymbol{G}, \boldsymbol{P}$ and $\boldsymbol{T}$ are prohibitive for a number of realistic decentralized stochastic control problems, the importance of approximate variants is clear.

3 An error-bounded heuristic search framework

The primary contribution of this section is an error-bounded heuristic search framework which builds upon approximate variants of greedy action-selection and accurate state-estimation operators. We also provide a provable bound on the error FB-HSVI algorithms would make by using these approximate operators instead of their exact counterparts. The result is a general algorithmic framework that allows for monitoring the divergence between the exact and approximate solutions of infinite-horizon and discounted decentralized stochastic control problems represented as Dec-POMDPs.

3.1 Error-bounded action-selection operators

This subsection characterizes error-bounded action-selection operators that select decentralized decision rules within α of maximizing the value.

Definition 6. *Let $\boldsymbol{\alpha} \in [0, \infty)^T$ be a real vector. An $\boldsymbol{\alpha}$-approximate action-selection operator $\widetilde{\boldsymbol{G}}\colon \mathcal{V} \mapsto (\triangle \mapsto \boldsymbol{A})$ is such that, at each time step $t \in \{0, 1, \ldots, T-1\}$, the decentralized decision rule found comes within $\boldsymbol{\alpha}(t)$ of maximizing the value:*

$$(\boldsymbol{T}_{(\boldsymbol{G}V_{\hat{M},\gamma,t+1})(\xi_t)}V_{\hat{M},\gamma,t+1})(\xi_t) - (\boldsymbol{T}_{(\widetilde{\boldsymbol{G}}V_{\hat{M},\gamma,t+1})(\xi_t)}V_{\hat{M},\gamma,t+1})(\xi_t) \leq \boldsymbol{\alpha}(t), \quad \forall \xi_t \in \triangle, V_{\hat{M},\gamma,t+1} \in \mathcal{V}.$$

For any positive T-dimensional vector $\boldsymbol{\alpha}$, a feature-based heuristic search value iteration, together with an $\boldsymbol{\alpha}$-approximate action-selection operator, terminates with a final estimate $V^{\boldsymbol{\alpha}}_{\hat{M},\gamma,0}(\xi_0)$. The error between this approximate value and the optimal value is bounded and the bound depends only upon parameter $\boldsymbol{\alpha}$ and γ.

Theorem 1. *The error introduced in FB-HSVI by using $\widetilde{\boldsymbol{G}}$ instead of $\boldsymbol{G}$ is bounded by $\sum_{t=0}^{T-1} \gamma^t \boldsymbol{\alpha}(t)$, assuming accurate estimation of the occupancy states during the planning phase. In particular, if $\boldsymbol{\alpha}(t) = \boldsymbol{\alpha}(t+1) = \ldots = \alpha$ for all $t \in \{0, 1, \ldots, T-1\}$, then the error is bounded by $\frac{1-\gamma^T}{1-\gamma}\alpha$.*

Proof. Let π^* and $\pi^{\boldsymbol{\alpha}}$ be decentralized policies that are optimal given that we use $(\boldsymbol{P}, \boldsymbol{G})$ and $(\boldsymbol{P}, \widetilde{\boldsymbol{G}})$, respectively. Vectors $\xi_1, \ldots, \xi_{T-1}$ being the occupancy states generated from

ξ_0 when applying π^*, it follows that:

$$
\begin{aligned}
& V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - V^{\pi^\alpha}_{\hat{M},\gamma,0}(\xi_0) \\
&= \left(\textstyle\sum_{t=0}^{T-1} \gamma^t \boldsymbol{R}(\xi_t, \pi^*_t)\right) - V^{\pi^\alpha}_{\hat{M},\gamma,0}(\xi_0) \quad \text{(definition of } V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0)\text{)}, \\
&= \left(\textstyle\sum_{t=0}^{T-1} \gamma^t \boldsymbol{R}(\xi_t, \pi^*_t)\right) + \textstyle\sum_{t=1}^{T-1} \left(\gamma^t V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t) - \gamma^t V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right) - V^{\pi^\alpha}_{\hat{M},\gamma,0}(\xi_0) \quad \text{(adding zero)}.
\end{aligned}
$$

Next, we use the fact that $V^{\pi^\alpha}_{\hat{M},\gamma,T}(\xi_T) = 0$ to re-arrange terms:

$$
\begin{aligned}
&= \left(\textstyle\sum_{t=0}^{T-1} \gamma^t \boldsymbol{R}(\xi_t, \pi^*_t)\right) + \left(\gamma^T V^{\pi^\alpha}_{\hat{M},\gamma,T}(\xi_T) + \textstyle\sum_{t=1}^{T-1} \gamma^t V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right) - \left(\gamma^0 V^{\pi^\alpha}_{\hat{M},\gamma,0}(\xi_0) + \textstyle\sum_{t=1}^{T-1} \gamma^t V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right), \\
&= \left(\textstyle\sum_{t=0}^{T-1} \gamma^t \boldsymbol{R}(\xi_t, \pi^*_t)\right) + \left(\textstyle\sum_{t=0}^{T-1} \gamma^{t+1} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1}(\boldsymbol{P}(\xi_t, \pi^*_t))\right) - \left(\textstyle\sum_{t=0}^{T-1} \gamma^t V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right), \\
&= \textstyle\sum_{t=0}^{T-1} \gamma^t \left(\boldsymbol{R}(\xi_t, \pi^*_t) + \gamma V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1}(\boldsymbol{P}(\xi_t, \pi^*_t)) - V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right), \\
&= \textstyle\sum_{t=0}^{T-1} \gamma^t \left(V^{\pi^*_t, \pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t}(\xi_t) - V^{\pi^\alpha_{t:T-1}}_{\hat{M},\gamma,t}(\xi_t)\right), \\
&= \textstyle\sum_{t=0}^{T-1} \gamma^t \left((\boldsymbol{T}_{\pi^*_t} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t) - (\boldsymbol{T}_{\pi^\alpha_t} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t)\right), \\
&\leq \textstyle\sum_{t=0}^{T-1} \gamma^t \left((\boldsymbol{T}_{(\boldsymbol{G} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t)} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t) - (\boldsymbol{T}_{(\widetilde{\boldsymbol{G}} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t)} V^{\pi^\alpha_{t+1:T-1}}_{\hat{M},\gamma,t+1})(\xi_t)\right), \\
&= \textstyle\sum_{t=0}^{T-1} \gamma^t \alpha(t).
\end{aligned}
$$

Thus, the error between $V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0)$ and $V^{\pi^\alpha}_{\hat{M},\gamma,0}(\xi_0)$ is bounded by $\sum_{t=0}^{T-1} \gamma^t \alpha(t)$. □

To the best of our knowledge, in decentralized stochastic control theory, this is the first attempt to monitor and bound the error made by using approximate action-selection instead of greedy action-selection. This bound comes with a natural interpretation: *all time steps are not equally relevant to the final error*. Indeed, due to discounted errors, approximate action-selection operators give more credit to errors they make at the earlier stages of the process. In other words, one can tolerate more approximation error at occupancy states that appear later in the process.

The problem of assigning errors to time steps goes beyond the scope of this paper, and will be addressed in the future. However, given the error vector $\boldsymbol{\alpha}$, another problem consists in finding a practical algorithm for selecting error-bounded actions over time steps. To do so, one can make use of the same branch-and-bound algorithms used for selecting greedy actions [7,8]. Except that, now, these algorithms need to be interrupted whenever the gap between lower and upper bounds is α. In that case, we are guaranteed the returned action has value within α of the optimal value, as targeted.

3.2 Error-bounded state-estimation operators: definition and example

This subsection discusses the long term behavior of successive applications of an approximate state-estimation operator. Next, we formally define the family of approximate state-estimation operators we target. Then, we exhibit one such operator. And finally, we derive theoretical guarantees.

Since we are interested in quantifying the error between occupancy states, we choose the *total variational distance* as a metric for measuring their distance. The total variational distance between two probability distributions ξ and ξ' on $[0,1]^{|S\|\Theta|}$ is defined

by $\|\xi - \xi'\|_{\text{TV}} = \frac{1}{2}\sum_{s\in S,\theta\in\Theta}|\xi(s,\theta) - \xi'(s,\theta)|, \quad \forall \xi, \xi' \in \triangle$. Informally, the total variational distance $\|\xi - \xi'\|_{\text{TV}}$ defines the minimal probability mass that would have to be re-assigned in order to transform occupancy state ξ into occupancy state ξ'. The following definition of approximate state-estimation operator $\widetilde{\boldsymbol{P}}_{\pi_t}$ guarantees that, for any occupancy state $\xi_t \in \triangle_t$, we have $\|\xi_t \boldsymbol{P}_{\pi_t} - \xi_t \widetilde{\boldsymbol{P}}_{\pi_t}\|_{\text{TV}} \leq \delta$.

Definition 7. *Let $\delta \in [0,1]$ be a small scalar. Then, for each decentralized decision rule $\pi_t \in \boldsymbol{A}_t$, transition matrix $\widetilde{\boldsymbol{P}}_{\pi_t}$ is a δ-approximation of $\boldsymbol{P}_{\pi_t}$ if, for any occupancy state $\xi_t \in \triangle_t$, there exists $\delta' \in [0,\delta]$ and $(\xi'_{t+1}, \xi''_{t+1}, \widetilde{\xi}''_{t+1}) \in \triangle^3_{t+1}$ such that*

$$\xi_t \boldsymbol{P}_{\pi_t} = (1-\delta')\xi'_{t+1} + \delta'\xi''_{t+1} \quad \text{and} \quad \xi_t \widetilde{\boldsymbol{P}}_{\pi_t} = (1-\delta')\xi'_{t+1} + \delta'\widetilde{\xi}''_{t+1}.$$

Now, we introduce and describe Algorithm 2 for constructing an artificial occupancy state that is within δ (in terms of variational distance) from the original occupancy state. To ensure the total variational distance between artificial and original occupancy states is upper bounded by δ, the algorithm clusters together private histories of the original occupancy state that are close enough (see Definition 8). Then, it replaces each such cluster with a unique private history in that cluster. Finally, this private history represents the cluster in the artificial occupancy state.

Algorithm 2: The occupancy state approximation algorithm (OSA)

```
function OSA(ξ_t, π_t, δ)
    ξ̃_{t+1} ← 0 and C ← LABELS(ξ_t, π_t, δ)
    foreach s ∈ S and c ∈ C do ξ̃_{t+1}(s,c) ← Σ_{θ∈[c]_{(ξ_t P_{π_t}, δ)}} ξ_t P_{π_t}(s,θ)
    return ξ̃_{t+1}

function LABELS(ξ_t, π_t, δ)
    foreach i ∈ {1,2,...,N} do
        C^i ← ∅ and Θ^i ← Θ^i(ξ_t P_{π_t})
        while Θ^i ≠ ∅ do
            c^i ← arg max_{θ^i∈Θ^i} |[θ^i]_{(ξ_t P_{π_t}, δ)}|
            C^i ← C^i ∪ {c^i} and Θ^i ← Θ^i \ [c^i]_{(ξ_t P_{π_t}, δ)}
    return ⊗_{i∈{1,2,...,N}} C^i
```

Before proceeding any further, we introduce the criterion we use, namely the *approximate probabilistic* measure.

Definition 8. *Let ξ_t be an occupancy state, and θ^i and $\bar{\theta}^i$ be two private histories in set $\Theta^i(\xi_t)$. We say that θ^i and $\bar{\theta}^i$ are δ-probabilistically close if and only if:*

$$\|Pr(X_t, Y_t|\xi_t, \theta^i) - Pr(X_t, Y_t|\xi_t, \bar{\theta}^i)\|_{TV} \leq \delta, \tag{9}$$

where X_t and Y_t denote random variables associated with states and other agent histories, respectively. We also denote $[\theta^i]_{(\xi_t,\delta)}$ the entire set of private histories $\bar{\theta}^i \in \Theta(\xi_t)$ that are δ-probabilistically close to θ^i and with respect to ξ_t.

By clustering private histories that are δ-probabilistically close with a single private history in that cluster, we produce (from Definition 8) an approximate occupancy state $\tilde{\xi}_t$ with respect to the original occupancy state ξ_t such that: $\|\xi_t - \tilde{\xi}_t\|_{\mathrm{TV}} \leq \delta$. Notice that Algorithm 2 is not guaranteed to produce an occupancy state with the minimum number of private histories. A more promising goal, which we do not address here, would be to find a clustering method that can identify the minimum number of clusters of private histories so that the total variational distance between original and artificial occupancy states is upper-bounded by δ.

3.3 Error-bounded state-estimation operators: theoretical analysis

We are now ready to bound the regret of using an approximate occupancy state instead of the accurate occupancy state. To do so, let $\boldsymbol{P}_{\pi_{0:t-1}} = \boldsymbol{P}_{\pi_0}\boldsymbol{P}_{\pi_1}\cdots\boldsymbol{P}_{\pi_{t-1}}$ for all time steps $t \in \{1, 2, \ldots, T\}$. Our analysis monitors the error we make step by step using approximate occupancy states.

Lemma 1. *The total variational distance between $\xi_0\widetilde{\boldsymbol{P}}_{\pi_{0:t-1}}$ and $\xi_0\boldsymbol{P}_{\pi_{0:t-1}}$ is bounded:*

$$\|\xi_0\widetilde{\boldsymbol{P}}_{\pi_{0:t-1}} - \xi_0\boldsymbol{P}_{\pi_{0:t-1}}\|_{TV} \leq 1 - (1-\delta)^t, \quad \forall t \in \{1, 2, \ldots, T\}. \tag{10}$$

Proof. The proof holds directly by expanding $\xi_0\widetilde{\boldsymbol{P}}_{\pi_{0:t}}$ and $\xi_0\boldsymbol{P}_{\pi_{0:t}}$ using Definition 7.

$$\begin{aligned}
&\|\xi_0\widetilde{\boldsymbol{P}}_{\pi_{0:t}} - \xi_0\boldsymbol{P}_{\pi_{0:t}}\|_{\mathrm{TV}},\\
&\leq \|(1-\delta)\xi_1'\boldsymbol{P}_{\pi_{1:t}} + \delta\tilde{\xi}_1''\widetilde{\boldsymbol{P}}_{\pi_{1:t}} - (1-\delta)\xi_1'\boldsymbol{P}_{\pi_{1:t}} - \delta\xi_1''\boldsymbol{P}_{\pi_{1:t}}\|_{\mathrm{TV}},\\
&\leq \|(1-\delta)^2\xi_2'\boldsymbol{P}_{\pi_{2:t}} + \delta(1-\delta)\tilde{\xi}_2''\widetilde{\boldsymbol{P}}_{\pi_{2:t}} + \delta\tilde{\xi}_1''\widetilde{\boldsymbol{P}}_{\pi_{1:t}} - (1-\delta)^2\xi_2'\boldsymbol{P}_{\pi_{2:t}} - \delta(1-\delta)\xi_2''\boldsymbol{P}_{\pi_{2:t}} - \delta\xi_1''\boldsymbol{P}_{\pi_{1:t}}\|_{\mathrm{TV}},\\
&\leq \|(1-\delta)^t\xi_t'\boldsymbol{P}_{\pi_{t:t}} + \textstyle\sum_{k=1}^{t}\delta(1-\delta)^{k-1}\tilde{\xi}_k''\boldsymbol{P}_{\pi_{k:t}} - (1-\delta)^t\xi_t'\boldsymbol{P}_{\pi_{t:t}} - \sum_{k=1}^{t}\delta(1-\delta)^{k-1}\xi_k''\boldsymbol{P}_{\pi_{k:t}}\|_{\mathrm{TV}},\\
&= \|\textstyle\sum_{k=1}^{t}\delta(1-\delta)^{k-1}\tilde{\xi}_k''\boldsymbol{P}_{\pi_{k:t}} - \sum_{k=1}^{t}\delta(1-\delta)^{k-1}\xi_k''\boldsymbol{P}_{\pi_{k:t}}\|_{\mathrm{TV}},\\
&\leq \textstyle\sum_{k=1}^{t}\delta(1-\delta)^{k-1}\|\tilde{\xi}_k''\boldsymbol{P}_{\pi_{k:t}} - \xi_k''\boldsymbol{P}_{\pi_{k:t}}\|_{\mathrm{TV}},\\
&\leq \textstyle\sum_{k=1}^{t}\delta(1-\delta)^{k-1},\\
&= 1-(1-\delta)^t. \qquad \square
\end{aligned}$$

It is worth noticing that approximation errors tend to increase exponentially as time goes on. The following derives the regret induced by approximating state estimates.

Theorem 2. *Let $\boldsymbol{\delta} \in [0,\infty)^T$ be a scalar vector. The error introduced in FB-HSVI by using a $\boldsymbol{\delta}$-approximate state-estimation operator instead of the exact state-estimation operator is bounded by $2\|r\|_\infty \sum_{t=0}^{T-1}\gamma^t[1 - \prod_{k=1}^{t}(1-\boldsymbol{\delta}(k))]$, assuming we use $\boldsymbol{G}$ for selecting decentralized decision rules. In particular, if $\boldsymbol{\delta}(t) = \delta$ for all time steps $t \in \{0, 1, \ldots, T-1\}$, then the error is bounded by $2\left(\frac{1-\gamma^T}{1-\gamma} - \frac{1-[\gamma(1-\delta)]^T}{1-\gamma(1-\delta)}\right)\|r\|_\infty$.*

Proof. Let π^* and $\tilde{\pi}$ be decentralized policies that are optimal given that we use accurate or approximate state-estimation operators, respectively. Define $\widetilde{V}^{\pi^*}_{\hat{M},\gamma,0}(\xi_0)$ as follows: $\widetilde{V}^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) = \sum_{t=0}^{T-1}\gamma^t\boldsymbol{R}(\xi_0\widetilde{\boldsymbol{P}}_{\pi^*_{0:t}}, \pi^*_t)$. Clearly, we have $\widetilde{V}^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) \leq V^{\tilde{\pi}}_{\hat{M},\gamma,0}(\xi_0)$ by definition of decentralized policy $\tilde{\pi}$. Using this property, we know that:

$$\begin{aligned}
V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - V^{\tilde{\pi}}_{\hat{M},\gamma,0}(\xi_0) &\leq V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - \widetilde{V}^{\pi^*}_{\hat{M},\gamma,0}(\xi_0),\\
&= \textstyle\sum_{t=0}^{T-1}\gamma^t\left(\boldsymbol{R}(\xi_0\boldsymbol{P}_{\pi^*_{0:t-1}}, \pi^*_t) - \boldsymbol{R}(\xi_0\widetilde{\boldsymbol{P}}_{\pi^*_{0:t-1}}, \pi^*_t)\right).
\end{aligned}$$

Since the value function is piecewise-linear and convex, $\boldsymbol{R}^{\pi_t^*} \equiv \boldsymbol{R}(\cdot, \pi_t^*)$ is a linear function of occupancy states. Thus, if we let $\langle \cdot, \cdot \rangle$ be an inner product, then we have

$$\begin{aligned}
V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - \widetilde{V}^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) &= \textstyle\sum_{t=0}^{T-1} \gamma^t \langle \boldsymbol{R}^{\pi_t^*}, \xi_0 \boldsymbol{P}_{\pi^*_{0:t-1}} - \xi_0 \widetilde{\boldsymbol{P}}_{\pi^*_{0:t-1}} \rangle, \quad \text{(by linearity of } \boldsymbol{R}^{\pi_t^*}) \\
&\leq \textstyle\sum_{t=0}^{T-1} \gamma^t \|\boldsymbol{R}^{\pi_t^*}\|_\infty \|\xi_0 \boldsymbol{P}_{\pi^*_{0:t-1}} - \xi_0 \widetilde{\boldsymbol{P}}_{\pi^*_{0:t-1}}\|_1, \quad \text{(Hölder's inequality)} \\
&\leq 2 \textstyle\sum_{t=0}^{T-1} \gamma^t \|\boldsymbol{R}^{\pi_t^*}\|_\infty \|\xi_0 \boldsymbol{P}_{\pi^*_{0:t-1}} - \xi_0 \widetilde{\boldsymbol{P}}_{\pi^*_{0:t-1}}\|_{\mathrm{TV}}, \quad (\text{where } \|x\|_1 = 2\|x\|_{\mathrm{TV}}) \\
&\leq 2\|r\|_\infty \textstyle\sum_{t=0}^{T-1} \gamma^t \|\xi_0 \boldsymbol{P}_{\pi^*_{0:t-1}} - \xi_0 \widetilde{\boldsymbol{P}}_{\pi^*_{0:t-1}}\|_{\mathrm{TV}}, \quad (\text{where } \|\boldsymbol{R}^{\pi_t^*}\|_\infty \leq \|r\|_\infty) \\
&\leq 2\|r\|_\infty \textstyle\sum_{t=0}^{T-1} \gamma^t \left[1 - \prod_{k=1}^{t} (1 - \boldsymbol{\delta}(k))\right].
\end{aligned}$$

This proves the result for any arbitrary $\boldsymbol{\delta} \in [0, \infty)^T$. If we let $\boldsymbol{\delta}(t) = \delta$ for all time step $t \in \{0, 1, \ldots, T-1\}$, then geometric series produce the following bound:

$$V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - V^{\tilde{\pi}}_{\hat{M},\gamma,0}(\xi_0) \leq 2\left(\tfrac{1-\gamma^T}{1-\gamma} - \tfrac{1-[\gamma(1-\delta)]^T}{1-\gamma(1-\delta)}\right)\|r\|_\infty,$$

which concludes the proof. □

Once again, in decentralized stochastic control settings, this is the first attempt to monitor and bound the error made by using approximate state-estimation operators. We note that, as time goes on, these operators become more tolerant to approximation errors. But there is no free lunch: approximation errors tend to increase as time goes on. This new bound provides a way to analyze this trade-off.

3.4 Convergence and error bounds

Given any arbitrary state-estimation and action-selection operators $\widetilde{\boldsymbol{P}}$ and $\widetilde{\boldsymbol{G}}$, which come with provable guarantees, the feature-based heuristic search value iteration produces an estimate $V_{\hat{M},\gamma,0}(\xi_0)$. The error between $V_{\hat{M},\gamma,0}(\xi_0)$ and the true value function $V^*_{\hat{M},\gamma,0}(\xi_0)$ is bounded. The error depends on quantities ϵ, $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$, each of which comes from a relaxation of the original problem. First, ϵ results from transforming an infinite horizon problem into a finite horizon one. Second, $\boldsymbol{\delta}$ represents the vector of errors the state-estimation operator allows at each time step. Finally, $\boldsymbol{\alpha}$ denotes the vector of errors the action-selection operator produces at each time step.

Theorem 3. *Let $\boldsymbol{\delta} \in [0, \infty)^T$ be the estimation operator parameter and $\boldsymbol{\alpha} \in [0, \infty)^T$ be the greedy operator parameter. The error of the feature-based heuristic search value iteration introduced by using $\widetilde{\boldsymbol{P}}$ and $\widetilde{\boldsymbol{G}}$ instead of $\boldsymbol{P}$ and $\boldsymbol{G}$ is bounded by:*

$$2\|r\|_\infty \sum_{t=0}^{T-1} \gamma^t \left[1 - \prod_{k=1}^{t} (1 - \boldsymbol{\delta}(k))\right] + \left(\sum_{t=0}^{T-1} \gamma^t \alpha(t)\right) + \varepsilon, \tag{11}$$

for any planning horizon $T = \lceil \log_\gamma ((1-\gamma)\varepsilon / \|r\|_\infty) \rceil$.

Proof. Let π^*, $\pi^{\boldsymbol{\alpha}}$ and $\pi^{\boldsymbol{\alpha},\boldsymbol{\delta}}$ be decentralized policies that are optimal given that we use $(\boldsymbol{P}, \boldsymbol{G})$, $(\boldsymbol{P}, \widetilde{\boldsymbol{G}})$ and $(\widetilde{\boldsymbol{P}}, \widetilde{\boldsymbol{G}})$, respectively. Then,

$$\begin{aligned}
V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - V^{\pi^{\boldsymbol{\alpha},\boldsymbol{\delta}}}_{\hat{M},\gamma,0}(\xi_0) &= \left(V^{\pi^*}_{\hat{M},\gamma,0}(\xi_0) - V^{\pi^{\boldsymbol{\alpha}}}_{\hat{M},\gamma,0}(\xi_0)\right) + \left(V^{\pi^{\boldsymbol{\alpha}}}_{\hat{M},\gamma,0}(\xi_0) - V^{\pi^{\boldsymbol{\alpha},\boldsymbol{\delta}}}_{\hat{M},\gamma,0}(\xi_0)\right), \\
&\leq 2\|r\|_\infty \textstyle\sum_{t=0}^{T-1} \gamma^t \left[1 - \prod_{k=1}^{t} (1 - \boldsymbol{\delta}(k))\right] + \left(\sum_{t=0}^{T-1} \gamma^t \alpha(t)\right).
\end{aligned} \tag{12}$$

This bound together with the fact that we search only for T-step policies is sufficient to demonstrate that the result holds. □

This theorem provides the first result quantifying the influence of different approximate operators in the overall performance of an algorithm for solving Dec-POMDPs. To the best of our knowledge, no similar results exist in Dec-POMDPs.

4 Experiments

This section presents experiments on a selection of infinite-horizon γ-discounted Dec-POMDPs including small-sized benchmarks (broadcast channel, multi-agent tiger, recycling robots and meeting in a 3x3 grid) and large-sized benchmarks (box-pushing, mars rover and wireless). For each benchmark, we ran the error-bounded feature-based heuristic search value iteration (EB-FB-HSVI) algorithm using parameters ϵ (pruning criterion), α (action-selection tolerance), and δ (state-estimation tolerance). Notice that, over the selection of benchmarks, action-selection tolerance α has only minor influence on performance results, so we set $\alpha = 0$ for many domains. We selected greedy actions using a constraint programming software, namely toulbar2 [11]. EB-FB-HSVI ran on a Mac OSX machine with 2.4GHz Dual-Core Intel and 2GB of RAM available.

Algorithm	$\lvert\Delta\rvert$	Time	$\underline{V}(\xi_0)$
Broadcast ($\lvert S\rvert = 4,\ \lvert A^i\rvert = 2,\ \lvert Z^i\rvert = 2$)			
FB-HSVI	102	**19.8s**	9.271
FB-HSVI($\delta = 0.01$)	435	**7.8s**	9.269
MPBVI	36	< 18000s	9.27
NLP	2	1s	9.1
Dec-tiger ($\lvert S\rvert = 2,\ \lvert A^i\rvert = 3,\ \lvert Z^i\rvert = 2$)			
FB-HSVI($\delta = 0.01$)	52	**6s**	13.448
FB-HSVI	25	**157.3s**	13.448
MPBVI	231	< 18000s	13.448
Peri	10×30	220s	13.45
PeriEM	7×10	6540s	9.42
Goal-directed	11	75s	5.04
Mealy NLP	4	29s	−1.49
EM	6	142s	−16.3
Recycling robots ($\lvert S\rvert = 4,\ \lvert A^i\rvert = 3,\ \lvert Z^i\rvert = 2$)			
FB-HSVI	109	**2.6s**	31.929
FB-HSVI($\delta = 0.01$)	108	**0s**	31.928
MPBVI	37	< 18000s	31.929
Mealy NLP	1	0s	31.928
Peri	6×30	77s	31.84
PeriEM	6×10	272s	31.80
EM	2	13s	31.50
IPG	4759	5918s	28.10
PI	15552	869s	27.20
Meeting in a 3x3 grid ($\lvert S\rvert = 81,\ \lvert A^i\rvert = 5,\ \lvert Z^i\rvert = 9$)			
FB-HSVI	108	**67s**	5.802
FB-HSVI($\delta = 0.01$)	88	**45s**	5.794
Peri	20×70	9714s	4.64

Algorithm	$\lvert\Delta\rvert$	Time	$\underline{V}(\xi_0)$
Box-pushing ($\lvert S\rvert = 100,\ \lvert A^i\rvert = 4,\ \lvert Z^i\rvert = 5$)			
FB-HSVI($\delta = 0.01$)	331	**1715.1s**	224.43
FB-HSVI($\alpha = 1, \delta = 0.05$)	288	**1405.7s**	224.26
FB-HSVI($\epsilon = 30$)	264	**15.24s**	199.42
MPBVI	305	> 18000s	224.12
Goal-directed	5	199s	149.85
Peri	15×30	5675s	148.65
Mealy NLP	4	774s	143.14
PeriEM	4×10	7164s	106.68
Mars rover ($\lvert S\rvert = 256,\ \lvert A^i\rvert = 6,\ \lvert Z^i\rvert = 8$)			
FB-HSVI($\delta = 0.01$)	136	**74.31s**	26.94
FB-HSVI($\alpha = 0.2$)	149	**85.72s**	26.92
FB-HSVI($\epsilon = 1$)	155	**32.5s**	26.77
Peri	10×30	6088s	24.13
Goal-directed	6	956s	21.48
Mealy NLP	3	396s	19.67
PeriEM	3×10	7132s	18.13
EM	3	5096s	17.75
Wireless ($\lvert S\rvert = 64,\ \lvert A^i\rvert = 2,\ \lvert Z^i\rvert = 6$)			
FB-HSVI($\delta = 0.01$)	897	**6309s**	−144.24
FB-HSVI($\alpha = 0.1$)	408	**6740s**	−140.37
FB-HSVI($\epsilon = 20$)	866	**6084s**	−176.59
MPBVI	374	> 18000s	−167.10
EM	3	6886s	−175.40
Peri	15×100	6492s	−181.24
PeriEM	2×10	3557s	−218.90
Mealy NLP	1	9s	−294.50

Table 1. Results for infinite-horizon decentralized POMDPs with $\gamma = 0.9$, and by default we set $\epsilon = 0.001$, $\alpha = 0$ and $\delta = 0$. Higher $\underline{V}(\xi_0)$ is better. Results for Mealy NLP, EM, PeriEM, PI, MPBVI and IPG were likely computed on different platforms, an therefore time comparisons may be approximate at best.

We compare EB-FB-HSVI for infinite-horizon Dec-POMDPs with state-of-the-art approximate and exact algorithms, including: optimal policy iteration (PI) [5]; incremental policy iteration (IPG) [2]; nonlinear programming (NLP and Mealy NLP) [1]; goal-directed algorithm [3]; periodic expectation maximization algorithm (EM, Peri and PeriEM) [16]; and modified point-based value iteration (MPBVI) [14]. Note that, while PI and IPG are optimal in theory, in practice they do not produce optimal solutions due to resources being exhausted before convergence. Table 1 reports performance results. For each domain and each algorithm, we report the lower-bound value function at the initial occupancy state $\underline{V}(\xi_0)$, the computation time required to achieve that value, and the memory requirement $|\underline{\Lambda}|$, which represents either the number of hyperplanes or the number of nodes in a policy graph.

In all tested benchmarks, EB-FB-HSVI achieves values higher or equal to the highest values that have been recorded so far, while being multiple orders of magnitude faster than state-of-the-art algorithms over many domains. In particular, over small-sized problems, EB-FB-HSVI demonstrates the best trade-off between the quality of the solution and the computation time. In addition, it is the only algorithm to provide provable bounds on the resulting solutions. In the broadcast channel, for example, both EB-FB-HSVI and MPBVI provide the highest value known so far, but EB-FB-HSVI comes with two advantages over MPBVI. First, it guarantees that value 9.271 is within 0.001 of the optimum. Second, it computed this value four orders of magnitude faster than MPBVI. Over large-sized problems, EB-FB-HSVI terminated with the highest values over all benchmarks for parameters $\alpha = 0, \delta = 0.01$ and $\epsilon = 0.001$. In the wireless problem, for example, the distance between the previous best value and EB-FB-HSVI's value is about 27, and EB-FB-HSVI was one order of magnitude faster than the previous best solver (MPBVI).

$\varepsilon_{apriori}$	$\varepsilon_{aposteriori}$	Gap(ξ_0)	$\varepsilon_{apriori}$	$\varepsilon_{aposteriori}$	Gap(ξ_0)
Broadcast			*Mars rover*		
1.651	0.018	0.003	16.62	0.773	0.83
Dec-tiger			*Box-pushing*		
166.7	0.727	0.001	16.74	1.8	4.99
Recycling robots			*Wireless*		
8.25	0.052	0.002	456.53	10.85	7.12

Table 2. Theoretical guarantees of EB-FB-HSVI($\delta = 0.01, \epsilon = 0.001$). We denote $\varepsilon_{apriori}$ the error computed a priori based on parameters δ and ϵ, and $\varepsilon_{aposteriori}$ the error computed a posteriori given approximation errors observed during the planning phase, both using Equation (11). Gap(ξ_0) = $\underline{V}(\xi_0) - \bar{V}(\xi_0)$, where $\underline{V}(\xi_0)$ is provided by FB-HSVI($\delta = 0.01, \epsilon = 0.001$) and $\bar{V}(\xi_0)$ results from EB-FB-HSVI(ϵ) for some ϵ.

We continue the study of the performance of EB-FB-HSVI with respect to tightness of error bounds. For each benchmark, we report in Table 2: a priori and a posteriori errors based on Equation (11) for FB-HSVI($\delta = 0.01, \epsilon = 0.001$); and gap Gap($\xi_0$) = $\underline{V}(\xi_0) - \bar{V}(\xi_0)$ based on FB-HSVI(ϵ). Notice that a posteriori errors were computed based on approximation errors observed during the planning phase. Overall, a posteriori errors are tighter than a priori errors and closer to gaps. The tightness of a posteriori error is mainly because the observed approximation errors were significantly smaller than the targeted ones. In the tiger problem, for example, the a priori error is about 166.7 whereas the a posteriori error and the gap are close: 0.727 and 0.001, re-

spectively. Surprisingly, in some domains such as mars rover and box-pushing, a posteriori errors are even smaller than gaps. This phenomenon occurs when EB-FB-HSVI(ϵ) exhausts the total available resources before convergence, i.e., the gap is larger than targeted error ϵ. The closeness between the gaps and the a posteriori errors demonstrate, at least over all tested domains, the tightness of our error bounds.

5 Discussion, conclusion and future work

This paper presented two relatively interdependent contributions towards error-bounded solutions for infinite-horizon discounted Dec-POMDPs. First, we introduce the first error-bounded algorithmic framework for monitoring and bounding the error we make by using approximate action-selection and state-estimation operators instead of their exact counterparts. Second, we extend the state-of-the-art algorithm for solving finite-horizon Dec-POMDPs, namely the feature-based heuristic search value iteration algorithm, to infinite-horizon discounted Dec-POMDPs. The major difference being that we can now use approximate operators instead of exact operators while still being able to provide theoretical guarantees on the quality of the resulting solution. Experimental results demonstrate that, when compared to state-of-the-art algorithms, the error-bounded feature-based heuristic search value iteration algorithm improves both values and computation times in many domains from the literature.

Though this paper provides the first attempts to monitor and bound the error made by using approximate operators in decentralized stochastic control, similar results exist in simpler settings. Such results can be traced back to max-norm-based analyses of value and policy iteration algorithms for γ-discounted MDPs [18], which prove that for some error α at each iteration there exists a stationary policy within $\frac{2\gamma}{(1-\gamma)^2}\alpha$ of the optimum. This result led to the development of much research on convergence arguments for γ-discounted MDPs and extensions including partially observable cases [17,21] and decentralized stochastic control settings [10]. Closer to our performance guarantees, [19,4] developed variations of value and policy iteration algorithms for computing non-stationary policies in γ-discounted MDPs for which the performance bounds can be significantly improved by a factor of $\frac{1}{1-\gamma}$. Hence, Theorem 1 can be viewed as an extension of [19] to decentralized stochastic control settings. However, Theorem 2 differs from previous performance bounds in many aspects. First, it is not derived from the max-norm analysis; instead we measure state-estimation errors we made steps by steps, which may result in tighter performance bounds. As a consequence, it does not fit within the standard scheme of performance bounds. Nonetheless, it allows us to accurately estimate errors made in practice on all tested benchmarks.

In the future, we plan to extend the feature-based heuristic search value iteration algorithm so as to learn how to dynamically assign approximation errors over time steps in order to minimize the total computation time while providing the targeted error bound. Another avenue we plan to follow, relies on how to automatically find the minimum number of clusters of private histories such that the artificial occupancy state based on clusters is within δ of the original occupancy state.

References

1. Amato, C., Bernstein, D., Zilberstein, S.: Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. Autonomous Agents and Multi-Agent Systems 21(3), 293–320 (2010)
2. Amato, C., Dibangoye, J., Zilberstein, S.: Incremental policy generation for finite-horizon Dec-POMDPs. In: ICAPS (2009), http://aaai.org/ocs/index.php/ICAPS/ICAPS09/paper/view/711/1086
3. Amato, C., Zilberstein, S.: Achieving goals in decentralized POMDPs. In: AAMAS (2009)
4. Bagnell, A., Kakade, S., Ng, A., Schneider, J.: Policy search by dynamic programming. In: NIPS. vol. 16 (2003)
5. Bernstein, D.S., Amato, C., Hansen, E.A., Zilberstein, S.: Policy iteration for decentralized control of Markov decision processes. Journal of Artificial Intelligence Research 34, 89–132 (2009)
6. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research 27(4), 819–840 (2002)
7. Dibangoye, J.S., Amato, C., Buffet, O., Charpillet, F.: Optimally solving Dec-POMDPs as continuous-state MDPs. In: IJCAI (2013)
8. Dibangoye, J.S., Amato, C., Buffet, O., Charpillet, F.: Optimally solving Dec-POMDPs as continuous-state MDPs: Theory and algorithms. Tech. Rep. RR-8517, Inria (April 2014)
9. Dibangoye, J.S., Mouaddib, A.I., Chai-draa, B.: Point-based incremental pruning heuristic for solving finite-horizon Dec-POMDPs. In: AAMAS (2009)
10. Dibangoye, J.S., Mouaddib, A.I., Chaib-draa, B.: Toward error-bounded algorithms for infinite-horizon Dec-POMDPs. In: AAMAS (2011)
11. de Givry, S., Heras, F., Zytnicki, M., Larrosa, J.: Existential arc consistency: Getting closer to full arc consistency in weighted CSPs. In: IJCAI (2005)
12. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: AAAI (2004)
13. Kumar, A., Zilberstein, S.: Point-based backup for decentralized POMDPs: Complexity and new algorithms. In: AAMAS (2010)
14. MacDermed, L.C., Isbell, C.: Point based value iteration with optimal belief compression for Dec-POMDPs. In: NIPS (2013)
15. Oliehoek, F.A., Spaan, M.T.J., Dibangoye, J.S., Amato, C.: Heuristic search for identical payoff Bayesian games. In: AAMAS (2010)
16. Pajarinen, J., Peltonen, J.: Periodic finite state controllers for efficient POMDP and DEC-POMDP planning. In: NIPS (2011)
17. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for POMDPs. In: IJCAI (2003)
18. Puterman, M.L.: Markov Decision Processes, Discrete Stochastic Dynamic Programming. Wiley-Interscience, Hoboken, New Jersey (1994)
19. Scherrer, B.: Improved and generalized upper bounds on the complexity of policy iteration. In: NIPS (2013)
20. Seuken, S., Zilberstein, S.: Formal models and algorithms for decentralized decision making under uncertainty. Autonomous Agents and Multi-Agent Systems 17(2), 190–250 (2008)
21. Smith, T., Simmons, R.G.: Point-based POMDP algorithms: Improved analysis and implementation. In: UAI (2005), http://dblp.uni-trier.de/db/conf/uai/uai2005.html#SmithS05